

Marianne van den Boomen
 Institute Media and Re/presentation
 University Utrecht
 Marianne.vandenBoomen@let.uu.nl

Indexicality: A matter of metaphors

*Conference The Ethics and Politics of Idexicality and Virtuality
 Bradford 1 July 2005*

Computer problems, though initially extremely puzzling and frustrating, may turn out to be ridiculously simple. To introduce some of the basic issues and problems of digital mediation I start with a story. It serves as a micro case study for my claims about metaphors and indexicality.

Years ago a friend of mine had problems with her brand new cable connection. She asked me to have a look at her computer. It was already on, and she clicked on the mail icon. 'See, no mail. And I'm sure I have mail, I forwarded some from my office. That man who did the installation said everything was working, just click and go. But well, no mail.'

'Okay', I said, 'Are you sure you are online?'

She pointed vaguely at the wires, still hanging all over the place, coming in from the hallway, swirling to her desk. 'I suppose so...'

I prayed these wires were okay, but however, that was not what I meant. I looked for the connection icon – fortunately, it was there, on her desktop, and moreover, it appeared to be correctly configured for her account. After a click it established the connection. And after clicking 'get mail' her mail came in promptly.

'You first have to connect with that icon', I said.

My friend was puzzled. 'Why, that telephone icon? I used that to dial up, but now I have a permanent connection. Just like I have at work - there I always see my mail immediately in the inbox, without having to click on anything.'

'Well,' I said, 'Apparently your computer at work is configured that way. That telephone icon stands for any kind of connection. But you can bypass it; it is possible to connect automatically when you start your computer or your mail program. And also to fetch your mail as soon as you start your mail program. We can arrange that, if you want.'

And so we did. After that my friend could forget that strange telephone icon.

1 Fooled by iconic literacy

That was all. It was in that respect a rare computer problem – mostly, such problems cost hours of trial and error and reconfiguration (or worse, re-installation or reformatting), but here we got a clear, comprehensible problem. Just one small conceptual error, just one small hidden step. Well, in

fact two of the same kind: the hidden step of connecting, and the hidden step of invoking the 'get mail' command.

Though this happened years ago, the story is still up to date. I think it is illustrative of how computerized mediation works, how it often fails to work, and how this mostly has to do with conceptual misreadings, misconfiguration and hidden steps in between.

The anecdote might too easily be read as a classic example of computer illiteracy, but that would miss the point. On the contrary, I would say, this is a story about literacy. My friend was no Internet newbie or computer illiterate. She was acquainted with e-mail at work and at home, she knew the difference between dialling up with a telephone modem and a direct connection, she was able to forward mail. In fact, she was too literate – especially in reading the icons on her desktop.

What exactly did happen? In fact she just took the iconic interface literally.

The icon of the mailbox was supposed to open her inbox, where her mail was supposed to be. Literally: mailbox = mail. The icon, the reference = the referent. The medium = the message. But in this case there were no messages. At least not until some steps in between were undertaken.

Let's take a closer look at these implied steps in between, the steps in between 'just click and go'. These can be analysed in terms of semiotic movements, or better, material-semiotic movements, since both human information processing/signifying, and system information processing are involved. My friend's problem can be seen as the result of the following three material-semiotic movements coming together.

1.1 Iconic metaphoric seduction

Firstly, the disposition to take the icon literally, take it for the referent: take the mailbox icon for the mail itself, take the telephone icon for using the phone itself. This is a phenomenon I propose to call 'icontology', the replacement of the icon with a specific ontological *state*.

This is not a matter of computer illiteracy, this is a matter of iconic metaphoric seduction by the interface. The inclination to take the icon for a specific state instead of a dynamic process¹ is very strong. In fact, this is part and parcel of the very function of icons. Icons are shorthand for complex machine processes. A desktop icon – mostly a visual representation of a metaphor, based on an associated resemblance or contiguity with the process involved - can be seen as an interface between complexity and simplicity, between executing machine processes and meaningful user action. That's the heart of iconic interfacing: translating complex processes into user-readable stable states, and translating user action into machine readable processes. The icon can only carry out it's signifying and executing job by concealing the involved complexities and highlighting stable states.

For example, the mailbox icon downplays the process it refers to: executing a particular mail program, including configurations of a particular Internet connection, a particular mail account and particular incoming and outgoing

¹ The distinction between process and state is elementary for programmers and software designers, but not for semioticians or social scientists. I owe this fruitful analytical insight to Bernhard Rieder, Paris-Utrecht Mini-Symposium November 2003.

mail servers. Instead, it highlights a specific result of the process (received mail) or a contiguous 'location' involved (the inbox). The same holds for the telephone icon. This icon conceals the fact that it is referring to executing an Internet connection program, including particular configurations of the hardware devices and the Internet account involved; it highlights instead the contiguous telephone device.²

o both icons refer indexically, causally to machine processes to be executed, and not to a specific state, place, or thing. But at the same time they conceal this, and represent certain ontological states as signifieds, in order to provide a clear human readable sign. We might say: The icons conceal their material dynamic reference by 'ontologising' the process involved into a stable representation.³ Or better, the metaphorical icons on our desktops do their work by *representing* an ontologised simplicity, and *depresenting* the dynamic complexity. This is the way the icon manages dynamic complexity, that's exactly the task we delegated to these icons. And that's why we are seduced, indeed compelled, to take icons literally, at 'interface value'.

1.2 The desire for immediacy

Secondly, there is the phenomenon of expected immediacy. My friend expected the mail to be in the inbox, immediately, just by looking there. In a certain sense, she was far ahead of her time; she was enacting the ideal of ubiquitous, invisible computing, which makes things happen just by looking or – even more utopian – just by thinking. These interfaceless computers have not yet arrived in our homes (and probably never will), but our contemporary interfaced computers do their utmost best to come near this ideal.

The suggestion of immediacy is already evoked by the shortcut icons themselves ('click-and-go'), but the desire for immediacy is also strongly sustained by a general Internet discourse full of 'one-click ideology': immediate access, instant messaging, one-click shopping, plug & play, automatic updates et cetera. These notions refer not so much to speed and the bridging of time and space, they refer to the ideal of the total transcendence of any speed, time and space. They refer to immediacy, which implies in the end the total erasure of reference itself. Reference and referent become one and the same, here and now, immediately. No mediation or transference in between; no material, temporal and spatial obstructions to conquer. These notions represent a desire and a belief in the possibility of total, direct, unmediated communication.

The discourse of the ideal of immediacy seems to be connected to the rise of 'new media', i.e. digital mediation, but in fact it is not that new. It has been

² The telephone icon in fact embodies in itself a forgotten replacement. We learned already not to take the telephone icon literally, since we 'know' it refers to the telephone modem and not the telephone device itself.

³ 'Ontologising' is not necessarily a movement of reification (to treat as a thing), neither of reduction (to absorb and dispense other concepts). Ontologising is a productive semiotic movement by which a process is translated into a state, a thing, or a space. These three ontologised instances need not necessarily be represented as isolated, as they can imply or evoke their own ontological dispositif. Ontologising (replacing dynamic processes with specific states, things, or places) is in a way the inverse twin of essentialising (replacing specific states, things, or places with presumed essential properties).

analysed as the motor behind the historically progressive sequence of all past and present new media, be it photography, film, or the Internet. Each time a new medium promised to make up for the mediation fallacies of a previous medium.⁴ Even deeply pessimistic studies of media ecology, arguing for cultural decline caused by the proliferation of new media, are rooted in the ideal of immediacy. These pessimistic analyses criticise new media as distorting the supposed immediacy of 'real', face-to-face communication.⁵ So whether one claims progress or decline by new media, the shared bottom line is the desire for immediacy. Immediacy is all around conceived as the best you can get, the standard by which any mediation and communication can be measured.

Hence, the desire for immediacy is the default disposition, but it only shows itself in a failure, in a break down which reveals retrospectively the expected immediacy. The fact that my friend expected the mail to be there immediately could only become clear because the mail just was *not* there. Had the mail indeed been there, she would probably not have noticed anything particular. The thousands of moments a day we might experience as immediacy never occur as such to our mind. Experiences of immediacy go by unnoticed, unperceived, unconscious, since there is no need for any reflection or second thoughts.

The idea of immediacy only shows itself when there is a sudden lapse in a taken for granted mediation. Immediacy can be seen as the degree zero of any mediation, a quest for a neutral, transparent mediation deprived of all traditional markers that heralds an encounter with media and interfaces as such, but which will only show itself as lack, as a gap between medium and message, between medium and mind, and between medium and reality.⁶

All media, in their own particular ways, have to deal with this gap, the ontological gap between on the one hand the material conditions and enactments of media, and on the other hand their representation, simulation or even production of reality. Contemporary desktop computers do this in their own way. They widen the gap by adding more and more complex processes in between the medium and the message – materialised in extra hardware devices, connection slots, applications, helper applications, software plug ins, patches, et cetera. And at the same time they diminish the gap by concealing these material processes: translating them into icons, shortcuts, embedded applications, automatic updates, system preconfigurations, et cetera.

Computers can be configured to automate all kinds of processes – connect to the Internet as soon as the mail program starts, get new mail as soon as the mail program runs - and when all this works properly, it may invoke a sense of seamless mediation, on the edge of immediacy. But to approach this state, a lot of invisible processing and labour has to be done. Labour by hardware machinery and energy, labour by software running, labour by acts of installation and configuration, labour by acts of choosing, clicking and typing. My friend was very aware of the hardware and wire installation labour which comes with a permanent Internet connection. She wisely delegated this to a professional. But she was not disposed to the configuration labour required by

⁴ Peters 1999, Bolter & Grusin 2000, De Vries, forthcoming.

⁵ See for instance Neil Postman, *Technopoly*. Penguin: New York 1993

⁶ The notion of degree zero is taken from Barthes 1953.

the software. Again, this is no instance of computer illiteracy, this is computer literacy, induced by frequent experience with computers. We tend to forget the labour of software - in fact, properly running software urges us to forget this, as this is the very reason we have delegated most of the processing labour to imperceivable software processes. The deal is: the software represents on the interface *that* it works, while derepresenting *how* it works. We are used to handling software as a black box, in which only the input and the output matters. This is in fact the paradigm of so called user-friendliness. Software only shows something of how it works or how it is supposed to work when it fails; then we need to open the black box, and consider all the steps and labour in between. In this sense, computer failures are analytically and educationally more illuminating than smoothly working computers.

1.3 Transference and transmediation

Thirdly, there is the phenomenon of transference between different domains. In the case of my friend, the transference was between home and work. Not only literally (forwarding mail from work to home) but also conceptually. My friend was used to the configuration of her computer at work, where she never had to connect to the Internet before receiving mail. She transferred this experience to her newly configured home system, based on her tacit knowledge of a few similar ontological states: both computers with a Windows operating system, both with a permanent Internet connection, both with a mail program installed. She implicitly assumed enough similarity between her office and her home computer to yield the same results. However, it was not enough, since transference of the results also implied transference of hidden configurations. The failed transference revealed a missing similarity between the two domains, which was needed to make the transference complete.

These kinds of transferences are ubiquitous in our contemporary thoroughly mediated and machinated environment. The mechanism of transference enables us to learn very easy to operate different devices and machines. Once you are acquainted with the operational concepts of switches and buttons on one device, you can transfer this knowledge to other devices. At least, your first trial is based on a transference of a previous experience. The technology of electricity and the electrification of society of course enlarged enormously the network of possible transference situations, but it is also a general learning principle, based on mimesis: the transference of experiences, acts and concepts from a familiar domain to a relatively new domain.

Computing in general relies heavily on this principle of transference. The ongoing streamlining of operating systems and interfaces – for example the standard menu tool bar with the standard items 'file', 'save', 'edit', 'copy' et cetera, but also standard keyboard combinations which work alike in different applications – assumes and reinforces the transference principle. Moreover, since computing is not only based on the technology of electricity but also on digitality, the network of possible transferences becomes even larger. While the above examples of transference occur on the level of operating the medium, i.e. the computer as such, transference can also occur between different media, on the level of the message itself: the digital-

material transport and translation of data and files. The message transference can be done between different computers (e.g. copying files via a disk, sending mail, FTP-downloading, web surfing), between different applications (e.g. converting a wav-file into a MP3-file, creating a PDF-file from a Word-document) and between different media modalities (e.g. scanning of a print picture, online streaming of television broadcasts).

Most of our daily computing activities consists of these acts of transference between different media modalities and between differently formatted messages. Sometimes these transmediations are manually enacted, most of the time they are mediated and enacted by hidden software processes. Most of the time these transfereces go on quite smoothless, but sometimes ruptures occur. Considering the numerous mutual interactions and interferences between different computers, different configurations, different applications and different formats, it is actually amazing how often medium and message transfereces succeed without failure.

In my friend's case it was indeed quite a miracle that only one little step in between was missing to make the transference from work to home complete. Think of the thousands of aspects that could possibly have been different: The man could have done a bad job with the wires or the configuration of the cable modem; her connection wizard at home could have been not yet configured; a network protocol or client could have been missing; her office computer could have been connected to a local area network, with quite another login procedure; her mail program at home could have been not properly configured for the mail server; she could have caught a vicious virus; a firewall could have been installed, et cetera.

It could have been so much worse - and we all know this by experience. Anyone who ever tried to copy the settings and configurations of an old computer to a new one, or tried to transfer the settings which work at your own computer to the computer of a friend, knows that a complete one-to-one transfer from one digital domain to another is practically impossible. Simply because two domains are never exactly the same. Small differences between the initial ontological states of the two domains may yield great differences in the process after transference. Transference from one domain to another can thus only be partial. Eventually the transference will be more a transformation or translation than an exact unmodified copy.

We know this, and still we keep forgetting this. Until some kind of rupture reminds us, and urges us to look for the differences between the presumed identical domains, and the implied hidden steps in between. These sudden ruptures in expected immediacy and taken for granted transference wake us up from our iconological slumber and urge us to peek into the black box of processing labour. But as soon as we have identified the hidden steps and fixed the problem, we can fall back into blissful forgetfulness. From then on we can afford to take the icons literally again, take them just 'at interface value' and not worry any more about their indexical reference to processing labour. And everything will work accordingly, blackboxed.

This process of learning and then forgetting again makes up computer literacy. It enables us to compute with a peace of mind. It is not my intention to disrupt completely this peace of mind, but a little more awareness of the

labour and coding which is going on inside the black box would be empowering. For ordinary users a slight transformation from the default state of iconological slumber to a state of 'stand by' would be already great. However, for researchers, scholars and designers in the field of new media the demands are higher. It is their duty to permanently open up and scrutinize the black boxes involved. My claim is that an analysis in terms of metaphors in action can provide the tools to do so.

2 Metaphors in action

In the above analysis of my friend's computer problem we can perceive metaphors in action on several levels.

First, desktop icons as such are shorthand symbols which stand for complex software procedures, and the way they represent particular applications is mostly based on an metaphorised resemblance or contiguity with the process involved. The mailbox icon is based on resemblance with postal mail; the connection icon is based on the contiguous device of the telephone (or better: the telephone modem, which indicates a double displacement/replacement). Second, we encountered the phenomenon of metaphoric seduction, or 'iconology', the disposition to take the icon literally - take the mailbox icon for the mail itself, take the telephone icon for using the phone itself. This disposition is on the one hand sustained by the two fold metaphorical labour of the icons: they *represent* at the screen interface an ontologised, metaphorised simplicity, while *depresenting* the dynamic complexity behind the screen. And on the other hand it is sustained by the human desire for immediacy ('click and go'). This desire for immediacy makes us all too willingly taking for granted that reference and referent, signifier and signified, collapse in one metaphorised sign: to click is to go, is to get the expected result. And thirdly, we saw that transferences between different domains were involved. Conceptual transferences (computer configurations from work to home) and material-mediated transferences (sending mail, computing in general). The transference of experiences, acts and concepts from a familiar domain to a relatively new domain is the basic paradigm of learning situations and experience, but it can also be analysed as paradigmatic of metaphor as such. The Greek etymological origins of the word are taken from *meta*, indicating 'after, over, across' (but also 'changed', which should be kept in mind) and *pherein* meaning 'to bear, to carry'. In modern Greek the word metaphor also means literally transport or transfer. Not coincidentally, buses in Athens are called 'metaphorai', as they are vehicles which transfer you from one place to another. Be they buses or words or visual icons, metaphors are media of transference.

2.1 Conceptual metaphor

Lakoff and Johnson's famous theory of metaphor⁷ is entirely based on this transference paradigm. In *Metaphors we live by* (1980) they formulate the principle loosely: 'The essence of metaphor is understanding and

⁷ Dubbed by Lakoff (1993) the Contemporary Theory of Metaphor (Lakoff 1993) but also known as the Cognitive Theory of Metaphor, and the Conceptual Theory of Metaphor.

experiencing one kind of thing in terms of another' (p 5) or 'to understand one domain in terms of another' (p 117). Later this is extended to a more formal definition: metaphors are defined as cross-domain mappings, sets of conceptual correspondences across conceptual domains, which can be mnemonically translated in the notational form of TARGET-DOMAIN IS SOURCE-DOMAIN (Lakoff 1993, p 207).

Classic examples of these conceptual metaphors are TIME IS MONEY ('wasting time', 'saving time') and ARGUMENT IS WAR ('defending a position', 'winning an argument'). It is important to notice that the mapping from source-domain to target-domain is never completely one-to-one. Some parts of the respective domains remain unused in the metaphorical transference, some parts are highlighted, others are downplayed. Metaphorical correspondence between source-domain and target-domain is always only partial.

In any case, transference of a concept from a source-domain to a target-domain is the basic mechanism of metaphor, according to this so-called Conceptual Theory of Metaphor. Note that metaphor here is not only metaphor as poetic figure; Lakoff and Johnson show extensively how metaphorical concepts are pervasive in daily life, and how they frame thoroughly our actions, thoughts and linguistic expressions, in a coherent though not always consistent system. They claim that metaphors are the building blocks of the whole human conceptual system, and as such grounded in sensory-motor experience (and perhaps even in neural processes).

Though their examples are all taken from language expressions, Lakoff and Johnson stress that metaphor is primarily a matter of thought, experience, and action, and only derivatively a matter of language (p 153). This suggests that their theory of metaphor is quite useful for the analysis of the material semiosis at stake in human-computer interaction mediated by, for instance, desktop icons.

We could map the source-domain and target-domain of the mailbox icon in this way. The underlying conceptual metaphor would be E-MAIL IS POSTAL MAIL, and the cross-domain mapping would look something like this:

SOURCE-DOMAIN	TARGET-DOMAIN
postal mail	e-mail
postbox	mail program
letters	messages
sending and receiving mail	push the send or get button
disposal of read mail	delete button
sorting mail	distribution to different folders
postal distribution system	mail-server at provider
delivery by postman	connecting to mail-server; invoking the fetch command

But here we encounter some problems. The last two correspondences are indeed at work in the transference but these are also exactly what is hidden in the iconic one-click metaphor. They *are* and *are not* part of the conceptual mapping. They are present in their absence. They are what I called *deprented*, while the other correspondences are represented.

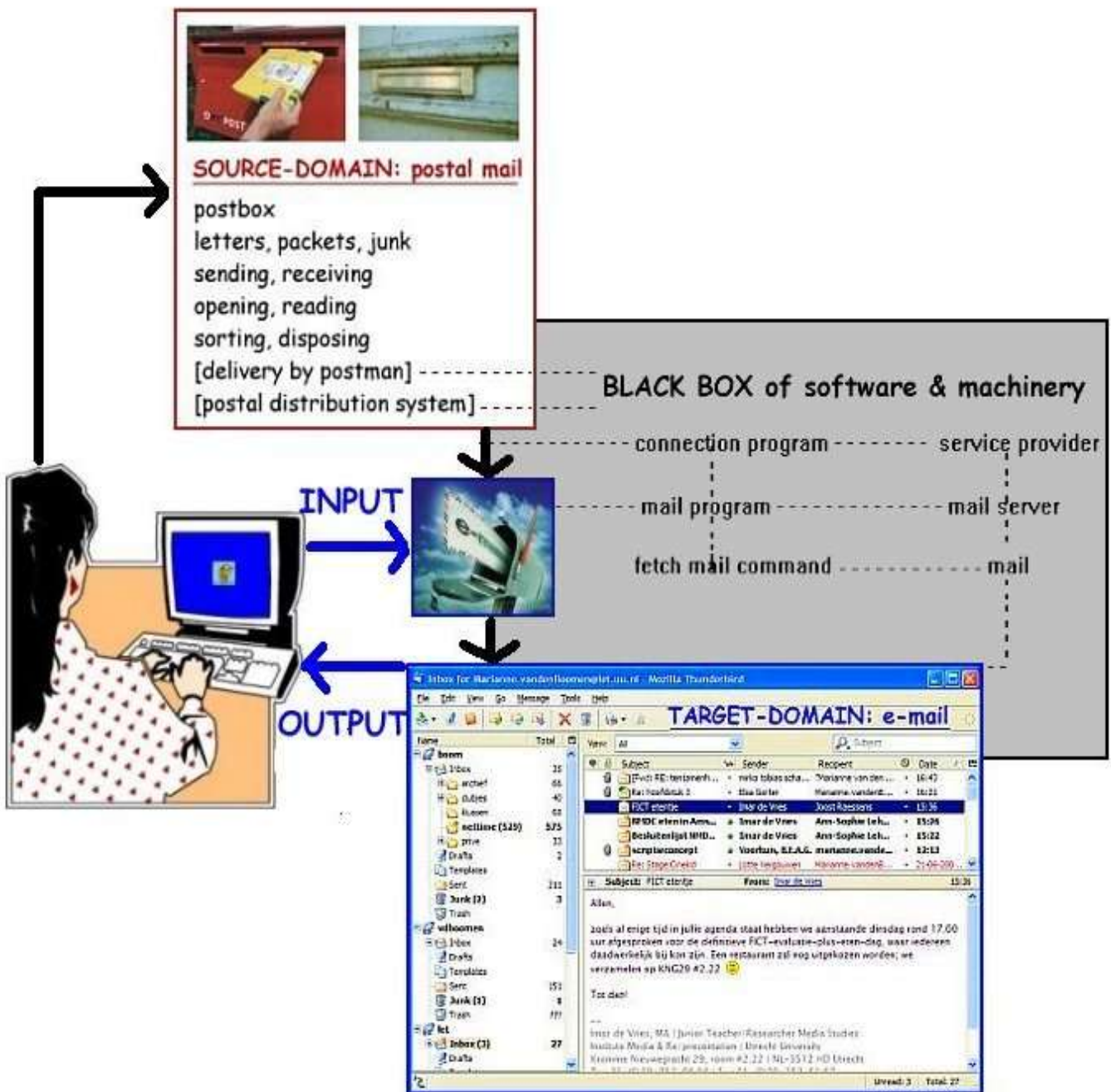
Note that this deprentation does not coincide with Lakoff and Johnson's notion of downplayed or hidden parts of the domains. Hidden or unused parts

in the metaphor E-MAIL IS POSTAL MAIL might consist of 'stamps' in the source domain, or 'viruses' in the target-domain. These aspects remain unused in the conceptual metaphor, they have no correspondence in the other domain, and they are not necessary to establish the metaphor. But the aspects of postal distribution and delivery in the source domain do have a correspondence in the target domain, and they are indispensable aspects of the metaphor, they lie at the heart of the success of the transference. The only thing in which these two aspects differ from the others is that they are not directly represented as operational for the user; their operation is delegated to hidden software processes. In other words, while these aspects are foundational for the metaphorical transference at stake, they have no proper place in the conceptual source and target-domains, since they are not operational in the conceptual human mind, but elsewhere. In the conceptual mind they operate at most as a negative, as a derepresentation of their elsewhere-ness.

Lakoff and Johnson's theory of metaphor and their method of mapping fails to give an account of this domain outside of the human mind. Since their theory is dubbed the Conceptual Theory of Metaphor, or alternatively the Cognitive Theory of Metaphor, this should come as no surprise. At the same time their focus on experiencing and acting suggests a possible break out of the mind, a connection to the material world. After all, they hold that metaphors emerge from experience and bodily interaction with phenomena in the world, that they can have material effects in the world, and that they can be 'made real in physical symptoms, social institutions, social practices, laws, and even foreign policy and forms of discourse and history' (Lakoff 1993, p 241). But in spite of this, their notion of metaphor is mentalistic, locked up inside the mind.

2.2 Material metaphor

Hence, the Conceptual Theory of Metaphor can not give an account of the metaphorical semiosis at work in human-computer interaction. What the theory of Lakoff and Johnson contributes to the understanding of the working of desktop icons, remains at the interface level. It does not go beyond the representations on the screen and the concepts in the mind of the user. It does not pay any attention to the material operations involved in the metaphorical transference, neither on the side of the user (clicking, configuring, or otherwise invoking software processes) nor on the side of the machine (code running, connections being made, fetching mail). In scheme we have a situation like this:



This is a situation of input and output, with a closed black box. The input and output mechanism consists of the metaphorical icon and the cross-domain mapping between source and target, but as long as the mapping does not involve the black box of software and machinery, the model in itself remains 'iconological'. It equates the reading and understanding of the icon (mediated by the source-domain) with the expected result (the target-domain). But reading and understanding the icon is not enough to let it do its job. While this suffices for linguistic or visual metaphors, it will not do for the desktop icon. In order for the desktop icon to function, it not only needs the conceptual

metaphorical grasp by the user, but also an operational grasp. The icon needs action to complete its referential task. Not only action from the user, who must click on the icon, but especially action inside the machine and the network. After all, a desktop icon refers indexically to machine processes to be executed. In that sense, the icon is a two-faced Janus: one side is headed towards the user, who must be able to read, understand and operate the icon, and the other side is headed towards software and machine processes. But the user is only able to read the icon as far as the other side is concealed and de-presented. That is, blackboxed.

What we need besides the notion of conceptual metaphor is a more material notion of metaphor, in order to open up that black box, and incorporate it in the analysis of desktop metaphors.

It may be that the notion of materiality sounds odd in the context of software and code. Is software material? Or is it non-material, virtual? The latter is in fact the dominant vision. Katherine Hayles showed in her classic work *How we became posthuman* (1999) how 'information has lost its body' during the history of informatics and cybernetics. She particularly revealed how much deliberate epistemological and ideological labour was involved to achieve this result. Indeed, software is nowadays usually conceived as disembodied, immaterial information, as a virtual pattern. In that sense, the virtual and the material are seen as oppositional dichotomy, echoing the age old Cartesian split between mind and matter.

Less Cartesian is a conception of software as ontologically situated somewhere between the virtual and the material. Though this may lead to a flexible hybrid ontology for software, it leaves intact the dichotomy of virtuality and materiality as such, and this is not very fruitful. The dichotomy takes no account of the fact that virtuality is always materially configured (by machinic mediatic processes and states) and the fact that materiality is always mediated (by perception and by representation). The relation between virtuality and materiality is further complicated by what I have dubbed processes of de-presentation: the semiotic productive concealment of material processes of mediation and transference.

What is at work in software can better be analysed in terms of 'material semiosis', as a practice of transference and cooperation (or conflict, or coexistence) between human information processing and machine information processing. This is the direction Katherine Hayles takes in her book *Writing machines* (2002). The book addresses (and expresses, in its typographical design) the materiality of analog and digital texts, and explores how literature has transformed itself from inscriptions rendered as durable marks of print to 'technotexts', dynamic images on computer screens. These dynamic images should not be conceived as classic visual representations formed of pixelated pictures or movies; Hayles focusses on the dynamic material qualities of 'literal' text: words, letters, design, typeface, and their new digital possibilities of interaction, iteration, interfacing and automation. In this context she introduces the concept of *material metaphor*, the instance of metaphor where the transfer does not take place between one word or concept and another, but between symbols and material apparatus. The material metaphor thus creates a dynamic connection between a symbol or

sign (more properly, a network of symbols/signs) and a physical artifact (Hayles 2002, p22). Material metaphors function as boundary objects, oscillating between operational interfaces and represented content, and as such are capable to transform both the interface and the content.

Hayles' case studies are all in the field of 'hypertextual art and design', and her insights on mediation and materiality do not come in a neat systematic theory or model, but they can be transferred to more mundane digital domains, such as simple mailbox icons. The concept of material metaphor provides a tool by which we can open the black box of software and machinery processing.

Compared to the notion of conceptual metaphor, which mediates between conceptual source and target-domains, the material metaphor focusses on another kind of transference and transmediation. It does not mediate between source and target, it mediates between force and event. While the conceptual metaphor can explain what happens inside the head of the user and on the surface of the screen, the material metaphor leads us from the hands of the user (the clicking action, i.e. force) through and behind the screen, inside the machine (events in the PC and on the Internet).

If we translate the situation at stake into the vocabulary of structuralist semiotics, we could say that the *signifier* (the mailbox icon) is connected to the *signified* (target-domain in Lakoff's terms, i.e. mail in inbox) by a double intervention of *reference*. The referent, the referred to object, should be distinguished from the sign,⁸ but should not be dismissed in this analysis. Its intervention is double, since there is both a *conceptual* reference (source-domain in Lakoff's terms, postal mail) and a *material* reference (events-domain, constituted by forces and causations in the blackboxed machinery and software processing).

It is tempting in this case to see the events-domain as a complete digital domain, consisting of digital code, floating ones and zeros – indeed, as Saussure's chain of differential signifiers come to live. Or better, a network of signifiers. But in fact the events-domain consists not only of digital stuff, i.e. software (client programs, server programs, computer languages, network protocols, commands, data packets) but also of hardware (PC's of the user, servers of Internet providers, cables, modems, routers, switches).

Unlike hardware devices we as humans have no direct access to the digital, for we are not able to make sense of bits of zeros and ones⁹, of naked signifiers without signifieds. But we do have partial access to the events domain. This is possible because of the intermediations and translations by material hardware, which can be represented to us as readable signs with implied signifieds. We are able to read and act upon parts of the software as

⁸ Reference is traditionally bracketed in structuralist inspired semiotics based on Saussure's linguistics. Saussure defines the sign as the arbitrary connection between signifier (material representation) and signified (mental concept). In this view the production of signifieds completely takes place in the differential interplay between the chain of signifiers, that is the language system. The referent, the object in the world to which the sign may refer to, lies outside of the domain of linguistics.

⁹ Though zeros and ones are already impossible to read for humans, the very notion of zeros and ones is already a translation to meet human conceptual capacities. In fact, machines don't 'read' zeros and ones, they just distinguish small voltage from no voltage, and operate upon these differences in current.

it is represented by configuration menu's ('mail server settings') or other interface buttons ('get mail', 'send mail'). And if we are suitably trained¹⁰ we are even able to read and modify the programme itself, not just its operation or configuration but its source code. And at yet another level we can have basic notions of software boundaries without needing any specific coding skills: we can hold in mind conceptual frames like 'connection programme', 'mail programme', 'service provider', 'mail server'. These are the concepts as designated in the black box scheme.

Unfortunately, Hayles' concept of material metaphor does not give an elaborate account of these level translations and their functioning in everyday computing. For an analysis of the way we think and act upon metaphorical interfaces we need a more specific and refined vocabulary, which is able to distinguish between different material-semiotic levels. In other words, what is needed beside the conceptual and the material focus is a more pragmatic approach.

3 Indexicality and virtuality

The philosopher and logician Charles Sanders Peirce (1839-1914) can be of great help here. Peirce is best known for his theory of signs ('semiotic') and his pragmatism, in short: relating belief to action instead of metaphysics. It is this focus on practical action and reasoning we need. Moreover, while a pragmatist perspective all too often implies an analytical bracketing of what's going on in the mind or what's going on in social discourse, Peirce's work escapes these pitfalls of blackboxing. Peircian pragmatism explicitly requires a theory of conversation, of social discourse and interaction, and his 'sign hypothesis' is designed as an alternative for the Cartesian notion of 'thought' as isolated, disembodied mental processes.¹¹

Peirce's theory of signs was in fact an ongoing work in progress – with several sequences of newly construed classifications - and as such never completely systematised in a whole theory. In any case his theory is a perpetual quest for the dynamics and possibilities of the sign: 'the essential function of a sign is to render inefficient relations efficient,- not to set them into action, but to establish a habit or general rule whereby they will act on occasion.' (Peirce, 1904, in Hardwick 1977)

3.1 Icon, index and symbol

Peirce's general rules always are derived from trichotomies, and Peirce brings several trichotomies into play. The most well known trichotomy is his distinction into three kinds of signs:

1. the *icon*, which exhibits a similarity or analogy to the subject of discourse
2. the *index*, which like a pronoun demonstrative or relative, forces the attention to the particular object intended without describing it
3. the *symbol*, the general name or description which signifies its object by means of an association of ideas or habitual connection between the name

¹⁰ And allowed to – for ordinary users this is only the case in open source software.

¹¹ Here I follow Anne Freadman (2004) who convincingly presents Peirce as the theorist of 'the machinery of talk' rather than of the mind and its contents.

an the character signified. (Peirce 1931)

This classification is based on the relation the sign has to its object. In short, these relations can be based on resemblance (iconical), on an existential or causal connection (indexical) or on convention (symbolical).

If we take metaphor as special instances of signs, we may wonder whether they are always iconic in the Peircian sense. Most theories of metaphor hold indeed that metaphor is a trope of resemblance, that metaphorical transference is based on a pre-existing resemblance between two different objects or concepts. However, Lakoff and Johnson in fact go beyond that with their more extended notion of correspondence. They hold that the perception of similarity can be *produced* by the conceptual metaphor. Besides, they conceive metonymy, synecdoche and symbols as special instances of metaphorical concepts, and these are grounded in other mechanisms than resemblance (Lakoff & Johnson 1980). Also, Katherine Hayles' material metaphor is in Peircian terms more indexical and symbolical than iconic: it is not based on a relation of resemblance but on a connection between symbol and material machinery.

Since metaphor can be analysed as conceiving sameness in two different phenomena, but also as conceiving two things at work in one thing, there is another problem regarding Peirce's trichotomy of signs based on a relation with the object. What is the *object* of a metaphorical sign, where should it be located: in the source-domain or in the target-domain? Or perhaps somewhere else?

Let's take the case of desktop icons, and relate the object to the source-domain. The *mailbox icon* would then be a genuine Peircian icon, since its relation to the source-domain is based on a particular resemblance, a similarity between postal mail and e-mail. But this does not hold for other desktop icons. For example, the *connection icon*, which depicts a telephone device connected to a computer, would be more an indexical sign than an icon, since its representation is not based on resemblance but on existential proximity, something contiguously involved in the causation chain. And the desktop icon for Microsoft's Internet Explorer, consisting of a logo in the form of a stylized 'e', (or the icon for Mozilla's Firefox browser, consisting of a fox curled around the globe) would be a Peircian symbolical sign, since it is based on a conventional relation with its object in the source-domain.

On the other hand, if we locate the object of the desktop icons not in the source-domain but in the target-domain, we get quite another classification. In that case all desktop icons are *indexical* signs. All desktop icons refer to existential, physical chains of causation, to machine processes to be executed, without describing these processes.

Hence, depending on where we locate the object of the desktop icons, we get different classifications of them as signs. Moreover, there is a third perspective on the sign and its object in which all desktop icons turn out to be *symbolical*. After all, we could say, no matter whether the representation the icon expresses is derived from similarity, contiguity or convention, they are all arbitrary and conventional. In order to function as desktop icons we have to learn what symbol stands for what application programme. And if we don't like the appearance of a particular default icon, we can replace it with another.

In any case, icons on our computer desktop are almost never icons in the Peircian sense. And still, at the same time, again regardless their particular representational strategy, they all exhibit what I have called 'icontology'. All desktop icons tend to equate and substitute the sign with its object of reference, to conceal its outward relations and dynamics, to act as if immediacy is within reach.

3.2 Firstness, Secondness and Thirdness

But Peirce's classification of signs has more levels than just the trichotomy of icon, index and symbol, based on the relation signifier-object. Peirce's other levels are based on material and formal categories of what he calls Firstness, Secondness, and Thirdness, and these categories on their turn also consist of trichotomies. Outlined in a chart Peirce's classification of signs shows the following levels (Cobley 1996, p.31):

		Ontological or material categories		
		<i>Firstness</i>	<i>Secondness</i>	<i>Thirdness</i>
Formal categories				
<i>Firstness</i>	A sign is:	quality QUALISIGN	actual existent SINSIGN	general law LEGISIGN
<i>Secondness</i>	A sign relates to its object:	resemblance ICON	existential relation INDEX	arbitrary, conventional SYMBOL
<i>Thirdness</i>	Its signified represents:	possibility RHEME	fact DICENT SIGN	reason ARGUMENT

According to this scheme a desktop icons may be analysed as Dicent ('factual') Legisign (expression of a general rule), which may relate to its object as Icon, Index or Symbol. In naming them Dicent Legisigns we foreground their symbolical and material-indexical reference. But we could also call them Dicent Sinsigns if we want to stress the double indexicality, or Rhematic Sinsigns or Iconic Legisigns to put more emphasis on the icontological mechanism. We may rumble around with Peirce's typology in several ways, in order to highlight different aspects of the signs at hand, and to make a particular argument about them. Peirce's classification can be read as a formal ontological system, but also as a rhetorical device, providing several dissecting tools.

In any case, it is important to stress that this classification should not be read as a static ontology. In Peirce's sign theory signifieds ('interpretants' in Peirce's terms) can become 'new' signifiers (signs), providing possible action in a chain or network of signifiers, producing new signs, with other indexical relations.¹² In the case of the desktop icons, these new signs may linger at the interface, but also inside the black box. With Peirce's classification we can try to track them down, by naming them, identifying them as signs, following their

¹² In that sense Peirce takes almost a Derridean non-metaphysical stance of deferral. Cf. Derrida 1972

movements, actions, forces and events.

This enables us to give an account of not only the signification at the conceptual level (i.e. the mechanism of iconology and derepresentation), but also of the material and indexical translations towards and inside the black box. Moreover, indexical translations sometimes creep out of the black box, and emerge on the interface, as coded symbols or analogical representations, for instance in the form of selection menu's, configuration settings or computer crashes. With Peirce's classification we can pin them down and analyse their 'thinghood' and 'eventhood'.

This also makes it possible to distinguish between the at least two levels of software processes inside the black box. I propose to distinguish them as two kinds of indexicality:

1. virtual indexicality, mediated by desktop icons or menu items as Rhematic Sinsigns, referring to the presence of installed application programmes on the computer, and the possibility to run them.
2. executing indexicality, a such non-represented on the interface, referring to the actual execution of the programme, the running itself. This executing process invokes a networked chain of other indexical forces and events on the PC and on Internet, each producing new kinds of virtual and executing indexicalities.

I'm not sure whether an ordinary user such as my friend would be helped with a user manual full of Peirce's trichotomies. But a Peircian inspired system manager or interface designer would be more imaginable. For new media scholars there is definitely work to do. Work which consists of analytical hacking. Analytical hacking, to open up the black boxes of code and machinery, of metaphor and event, and of virtuality and indexicality. In any case, it should be clear that there is no such thing as 'the loss of indexicality in the digital era', as claimed by many new media scholars. On the contrary, indexicality informs and constructs our world and our mind. And we need Peirce to trace down the virtual and executing indexicalities inside the machinery and the way they come back to us.

Bibliography

- Barthes, Roland. *De nulgraad van het schrijven*. (orig. 1953) Amsterdam: Meulenhoff 1970
- Bolter, Jay David & Grusin, Richard. *Remediation: Understanding new media*. Cambridge, MA: MIT Press 2000
- De Saussure, Ferdinand 'The object of linguistics'. In: *Course in general linguistics*. Bally, Charles (ed); Sechehaye, Albert (trans.) Glasgow: Fontana 1974.
- Derrida, Jacques. 'Signature event context'. In: *Limited Inc*. Evanston: Northwestern University Press 1972
- Freadman, Anne. *The machinery of talk: Charles Peirce and the sign hypothesis*. Stanford: Stanford University Press 2004
- Hardwick, Charles S. *Semiotic and sigifics: The correspondence between Charles S. Peirce and Victoria Lady Welby*. Bloomington: Indiana University Press 1977
- Hayles, N. Katherine. *How we became posthuman. Virtual bodies in cybernetics, literature and informatics*. Chigago: The University of Chigago Press 1999
- Hayles, N. Katherine. *Writing machines*. Cambridge, Massachusetts: MIT Press 2002

- Hayles, N. Katherine, 'Deeper into the machine: Learning to speak digital'. In *Computers and Composition* 19 (4) December 2002
- Lakoff, George. *Women, fire and dangerous things: What categories reveal about the mind*. Chicago: The University of Chicago Press 1987
- Lakoff, George 'The contemporary theory of metaphor'. In: Ortony, Andrew (ed.) *Metaphor and thought*. Cambridge, Cambridge University Press. 1993.
- Lakoff, George & Johnson, Mark. *Metaphors we live by*. Chicago: The University of Chicago Press 1980
- Lucy, Niall. *Beyond semiotics: Text, culture and technology*. London, New York: Continuum 2001
- Manovich, Lev. *The language of new media*. Cambridge, Massachusetts: MIT Press 2001
- Ortony, Andrew. (ed.). *Metaphor and thought*. (Second edition) Cambridge: Cambridge University Press 1993.
- Peirce, Charles Sanders. (ed. C. Hartshorne & P. Weiss) *Collected papers of Charles Sanders Peirce*. Cambridge, Mass: Harvard University Press 1931-58
- Peters, John Durham. *Speaking into the air: A history of the idea of communication*. Chicago: University of Chicago Press 1999
- Ricoeur, Paul. *The rule of metaphor: The creation of meaning in language*. New York: Routledge 1975
- Thwaites, Tony, Davis, Lloyd, & Mules, Warwick. *Introducing cultural and media studies: A semiotic approach*. Hampshire, New York: Palgrave 2002